Ekstraksi Teks Pada Halaman Website Renungan Rohani Menggunakan HTML Agility Pack

James Wijaya, *Departemen Sistem Informasi, Universitas Ciputra Surabaya, Indonesia*

***Abstrak*—Dengan adanya perkembangan teknologi informasi, orang-orang dapat mengakses berbagai informasi dari berbagai halaman web dengan menggunakan internet. Web Santapan Rohani adalah salah satu contoh website yang dapat digunakan oleh orang-orang terlebih khusus umat Kristiani untuk membaca renungan harian atau untuk melakukan saat teduh. Penelitian ini bertujuan menciptakan suatu teknologi ekstraksi informasi dari web Santapan Rohani yang berisikan renungan harian sehingga dapat membantu untuk analisa bagi penelitian-penelitian berikutnya yang dapat dikembangkan dari kehadiran teknologi ini. Halaman web memiliki bentuk yang semi-structured dan berisikan informasi berupa teks, gambar, video, URL, dan sebagainya. Hal ini menjadi kendala untuk dapat melakukan ekstraksi informasi dari halaman web. HTML Agility Pack merupakan salah satu tools terbaik yang dapat digunakan untuk melakukan HTML Parser dari suatu halaman web. Dengan menggunakan HTML Agility Pack dapat mempermudah untuk melakukan ekstraksi informasi dari berbagai halaman web, terlebih khusus untuk melakukan ekstraksi informasi pada renungan harian dari Web Santapan Rohani**

***Kata Kunci*—Ekstraksi Informasi, Halaman Website, Html Agility Pack, Renungan Rohani.**

I. Pendahuluan

Perkembangan teknologi komunikasi dan informasi yang semakin pesat dari hari ke hari semakin mempermudah kehidupan manusia. Dengan adanya koneksi internet dan pemanfaatannya yang sangat luas, manusia dapat mengakses berbagai informasi yang dibutuhkan dari sebuah website atau situs online dengan cepat. Suatu website tidak hanya berisikan informasi berbentuk teks, tetapi juga dapat berupa gambar, video, alamat URL, dan lain sebagainya.

Penelitian ini akan membahas mengenai ekstraksi informasi berupa teks dari sebuah halaman webiste renungan harian yang ditulis dalam Bahasa Indonesia. Website renungan harian yang akan digunakan adalah website dari Santapan Rohani. Santapan Rohani merupakan nama untuk publikasi renungan terjemahan berbahasa Indonesia dari *Our Daily Bread* (sebelumnya RBC). Santapan Rohani tersedia dalam bentuk cetak maupun online. Alamat situs online atau website dari Santapan Rohani adalah http://santapanrohani.org.

Website Santapan Rohani selain memiliki bacaan renungan harian sebagai konten utama, juga memiliki konten-konten lain, sehingga perlu dilakukan ekstraksi konten atau teks yang dibutuhkan dari halaman webiste tersebut. Fokus dari penelitian ini adalah melakukan ekstraksi teks dari website Santapan Rohani sehingga mendapatkan konten utama yang dibutuhkan. Konten utama yang didapatkan berupa kategori atau topik renungan, judul, ayat bacaan renungan harian, ayat emas atau utama, isi renungan, refleksi, dan kutipan inspirasi yang dapat digunakan dalam penelitian-penelitian selanjutnya yang berhubungan dengan bidang *text mining*.

Penelitian ini akan melakukan ekstraksi teks pada halaman website yang memiliki bentuk *semi*-structured (dalam format HTML). Dalam mempermudah melakukan proses ekstraksi teks maka dimanfaatkan sebuah *tools* dalam bentuk *library* (.dll) yang dapat membantu melakukan *parsing* dari format HTML yang didapat, yaitu HTML Agility Pack.

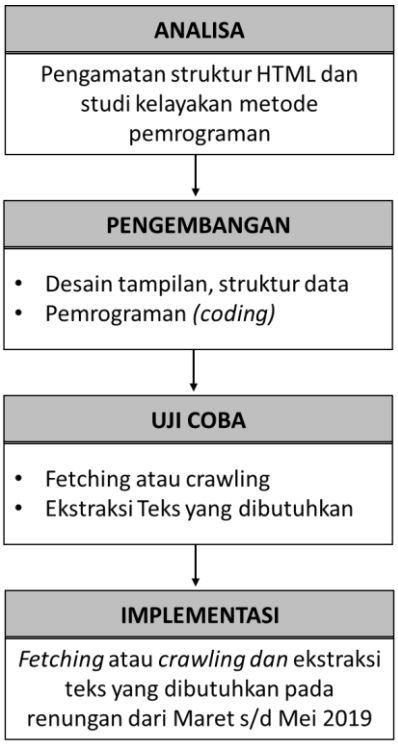
II. Metodologi Penelitian

Dalam penelitian ini dilakukan berbagai pengamatan untuk menentukan metode yang tepat dalam mengembangkan sebuah aplikasi ekstraksi teks dari halaman website. Metode *System Development Life Cycle* (SDLC) diterapkan pada penelitian ini, dimulai dari tahapan analisa, *development*, uji coba dan implementasi. Pada Gambar 1 merupakan ringkasan siklus pengembangan program yang diterapkan dalam penelitian ini.

Pada tahapan analisa, peneliti melakukan pengamatan terhadap halaman dari website Santapan Rohani untuk mendapatkan fakta-fakta yang dibutuhkan sebelum memulai pengembangan sistem ekstraksi teks (desain dan pemrograman) dan studi kelayakan mengenai metode pemrograman yang akan digunakan untuk membangun sistem ekstraksi teks ini. Setelah selesai mengembangkan sistem ekstraksi teks pada halaman website renungan harian Santapan Rohani dilakukan uji coba terhadap salah satu halaman konten renungan berdasarkan tanggal tertentu. Pengujian dilakukan beberapa kali untuk mendapatkan keyakinan bahwa program yang dikembangkan selama proses pemrograman (*coding*) telah dapat berjalan dengan baik. Implementasi dari program yang telah dikembangkan digunakan untuk melakukan *fetching* atau *crawling* dan ekstraksi teks yang dibutuhkan pada sejumlah halaman

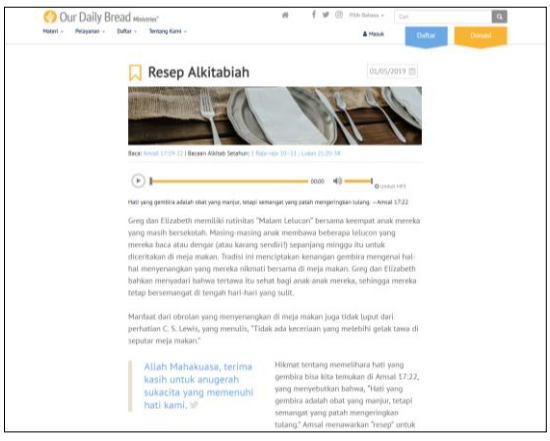
James Wijaya, Departemen Sistem Informasi, Universitas Ciputra Surabaya, Surabaya, Jawa Timur, Indonesia (e-mail: james.wijaya@ciputra.ac.id)

renungan harian yang diterbitkan pada bulan Maret s/d Mei 2019.



Gambar. 1. Siklus Pengembangan Program

Pada Gambar 2 merupakan potongan tampilan dari halaman konten renungan Santapan Rohani. Pada halaman konten renungan ini terdapat konten berupa teks maupun multimedia (suara). Selain itu, halaman ini terdiri dari beberapa bagian, yaitu: atas (*header*), tengah (*content*), dan bawah (*footer*). Pada bagian *header* terdiri dari logo/*brand*, kotak pencarian, *link* media sosial, dan menu, sedangkan pada bagian *content* memiliki renungan dalam bentuk teks (terdiri dari judul, informasi ayat bacaan, kutipan ayat atau ayat emas, isi renungan, bagian refleksi atau doa, kutipan inspirasi, nama penulis) maupun renungan dalam bentuk audio. Bagian *content* inilah yang menjadi target penelitian untuk melakukan ekstraksi teks.



Gambar. 2. Halaman Konten Renungan dari Santapan Rohani

Metode dari penelitian ini dimulai dari mengamati format URL yang dimiliki oleh website Santapan Rohani, pengamatan struktur HTML dari halaman website, pengambilan seluruh konten halaman dalam bentuk HTML (*crawling* atau *fetching*), preproses dan ekstraksi teks dari format halaman website (dalam bentuk HTML), dan pemberian label (*labelling*) terhadap konten-konten utama yang dibutuhkan (yaitu: kategori atau topik renungan, judul renungan, ayat bacaan renungan, ayat emas atau utama, isi renungan, refleksi, dan kutipan inspirasi).

A. Santapan Rohani

Berdasarkan pencarian informasi pada website Santapan Rohani (http://santapanrohani.org), Santapan Rohani adalah salah satu judul buku renungan rohani bagi umat Kristiani dalam Bahasa Indonesia yang diterbitkan setiap tiga bulan sekali. Selain tersedia dalam bentuk cetak, Santapan Rohani juga bisa diakses melalui situs website. Buku Santapan Rohani juga ditulis dalam Bahasa Inggris dan Mandarin (Gambar 3).

Gambar. 3. Versi Cetak Santapan Rohani dalam Berbagai Bahasa

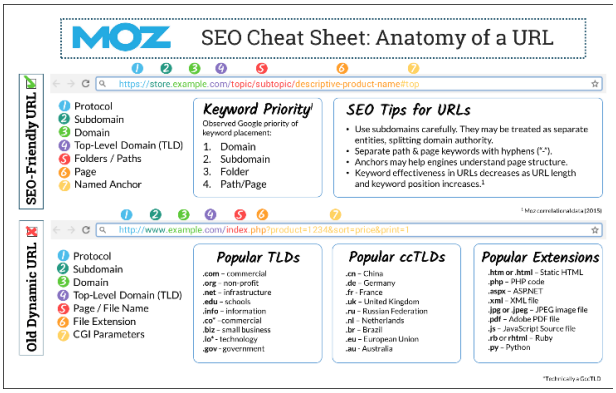
Santapan Rohani merupakan buku renungan terjemahan dari *Our Daily Bread* yang diterbitkan oleh Our Daily Bread Ministries. Our Daily Bread Ministries merupakan lembaga pelayanan interdenominasi nirlaba yang menyediakan materi-materi penuntun dalam berbagai media, seperti: siaran radio atau televisi, DVD, siaran podcast, buku, situs Internet, media sosial, atau aplikasi mobile. Materi ini ditujukan bagi setiap pribadi yang ingin mengalami pertumbuhan hubungan pribadi dengan Tuhan. Santapan Rohani memiliki format yang sederhana dan pada setiap artikel renungan yang disajikan terdapat ayat Alkitab, kisah inspiratif, dan petunjuk penerapannya dalam kehidupan sehari-hari. [1]

B. SEO Friendly URL

URL adalah singkatan dari *Uniform Resource Locator*, sedangkan SEO adalah singkatan dari *Search Engine Optimization*. Suatu URL digunakan untuk mengubah IP Address yang digunakan komputer untuk berkomunikasi dengan server menjadi sebuah teks yang dapat dikenali atau dibaca oleh manusia atau yang dikenal dengan sebutan nama domain (*domain name*) [2].

Search Engine Optimization (SEO) bertujuan agar sebuah situs dapat berada pada posisi teratas pada hasil pencarian berdasarkan kata kunci tertentu yang ditargetkan, sehingga situs tersebut dapat memiliki peluang besar untuk mendapat pengunjung [3]. Salah satu cara untuk meningkatkan peringkatkan sebuah website dengan pemanfaatan *SEO*

Friendly URL. Struktur *SEO Friendly URL* merupakan bentuk atau format yang konsisten dari sebuah URL yang mudah dibaca oleh manusia dan mesin pencari [4].



Gambar. 4. Struktur Anatomi dari URL

Struktur anatomi dari sebuah URL ditunjukkan pada Gambar 4 yang bersumber dari http://moz.com [2]. URL terdiri dari 2 macam, yaitu: *Old Dynamic URL*, dan *SEO Friendly URL*. Secara umum, URL terdiri dari: protokol (http atau https), subdomain, domain, *Top-Level Domain* (TLD). Namun, terdapat perbedaan antara kedua URL dimulai dari bagian 5 hingga 7 seperti yang diilustrasikan pada gambar. *Old Dynamic URL* adalah pemodelan URL yang masih memanfaatkan nama file (bagian 5 dan 6 dari *Old Dynamic URL* pada Gambar 4) dan parameter yang dibutuhkan untuk menghasilkan tampilan. Contoh parameter yang dimaksud adalah "?product=1234&sort=price&print=1", dapat dilihat pada bagian 7. Struktur dari *Old Dynamic URL* tidak dapat dikenali dengan mudah oleh manusia untuk dipahami secara langsung maupun untuk mesin pencarian (*Search Engine*).

Struktur dari *SEO-Friendly URL* sangat memudahkan untuk dibaca dan dikenali baik oleh manusia maupun mesin pencarian. Bagian 5 pada *SEO-Friendly URL* menunjukkan nama folder atau *path* dari halaman yang akan dituju (bagian 6). Bagian 7 merupakan *section* tertentu yang ingin langsung ditampilkan atau dituju dari suatu halaman, misal: *section* top.

Website Santapan Rohani menggunakan format *SEO Friendly URL* sehingga dapat mempermudah pembacaan format URL untuk kemudian dilakukan pengambilan seluruh konten halaman dalam bentuk HTML. Format URL dari Santapan Rohani memiliki pola yang disusun dalam format secara berurut, yaitu, nama website (http://santapanrohani.org) diikuti dengan tahun, bulan, tanggal, dan judul renungan. Berikut ini adalah contoh formal URL dari Santapan Rohani (http://santapanrohani.org/2019/01/05/diubahkan-dan-mengubahkan/) untuk renungan harian pada 5 Januari 2019 dengan judul "Diubahkan dan Mengubahkan".

C. *HTML dan DOM*

Hypertext Markup Language (HTML) merupakan bahasa standar yang digunakan pengembangan sebuah aplikasi berbasis website [5]. Struktur HTML ditandai dengan penggunaan tag <>, di mana sebuah konten akan dibungkus oleh tag pembuka dan tag penutup, seperti yang ditunjukkan pada Segmen Program 1.

Document Object Model (DOM) adalah sebuah standar yang ditetapkan oleh W3C (*World Wide Web Consortium*) untuk mengakses sebuah dokumen. Dengan menggunakan DOM, program atau *script* dapat mengakses dan memperbaharui konten, struktur, dan *style* dokumen dengan lebih dinamis [6]. Terdapat 3 jenis DOM, yaitu: (1) Core DOM untuk standar pemodelan bagi semua jenis dokumen, (2) XML DOM untuk standar pemodelan bagi dokumen XML, (3) HTML DOM untuk standar pemodelan bagi dokumen HTML [7].

Segmen Program 1. Struktur HTML

```
<TABLE>
  <TR>
    <TD>Grove</TD>
    <TD>Aeolian</TD>
  </TR>
  <TR>
    <TD>Charlie</TD>
    <TD>Dorian</TD>
  </TR>
</TABLE>
```

Tag <TABLE> digunakan untuk menampilkan sebuah tabel, di mana banyaknya tag <TR> menentukan banyaknya baris yang akan dihasilkan, dan tag <TD> menentukan banyaknya kolom yang akan dihasilkan. Pada merupakan contoh dari struktur HTML untuk menampilkan sebuah tabel dengan 2 baris, dan 2 kolom untuk setiap barisnya. Pada baris ke-1, secara berurut kolom-kolomnya berisi Groove (baris ke-1, kolom ke-1) dan Aeolian (baris ke-1, kolom ke-2), sedangkan baris ke-2 kolom-kolomnya berisi Charlie (baris ke-2, kolom ke-1) dan Dorian (baris ke-2, kolom ke-2).

Pada Segmen Program 2 merupakan ilustrasi HTML dari halaman renungan harian Santapan Rohani. Judul Renungan didapatkan dengan melakukan *parsing* HTML dari konten pada tag <title> atau tag <h2> yang memiliki *class* bernama "entry-title", sedangkan untuk Ayat Bacaan Renungan, Ayat Emas, Isi Renungan, Refleksi, dan Kutipan Inspirasi secara berurut dari *class* bernama "passage-box", "verse-box", "post-content", "poem-box", dan "thought-box".

Segmen Program 2. Ilustrasi Halaman HTML dari Santapan Rohani

```
<html>
  <head>
    <title>Strategi Terbaik</title>
  </head>
  <body>
    <h2 class="entry-title"> Strategi Terbaik
    </h2>
    <div class="passage-box">...</div>
    <div class="verse-box">...</div>
    <div class="post-content">...</div>
    <div class="poem-box">...</div>
    <div class="thought-box">...</div>
  </body>
</html>
```

D. *HTML Agility Pack*

HTML Agility Pack merupakan salah satu HTML Parser yang dapat digunakan untuk menulis atau membaca DOM (*Document Object Model*) [8]. HTML Agility Pack berbentuk *library* (.dll) berbasis .NET dan dapat membantu untuk melakukan *parsing* file HTML [9]. Dokumentasi dari

penggunaan HTML Agility Pack lebih lanjut dapat dilihat pada https://html-agility-pack.net/documentation.

Fungsi-fungsi yang akan digunakan pada penelitian ini dari HTML Agility Pack dalam penelitian ini adalah fungsi HTML Selectors yang berfungsi untuk memilih *node* HTML dari sebuah dokumen atua struktur HTML. Terdapat 2 fungsi dari HTML Selectors, yaitu SelectNodes() untuk memilih sekumpulan node, dan SelectSingleNode() untuk memilih node pertama yang sesuai dengan ekspresi X-Path yang diberikan pada parameter dari kedua fungsi tersebut.

```
1  // @nuget: HtmlAgilityPack
2
3  using System;
4  using HtmlAgilityPack;
5
6  public class Program
7  {
8      public static void Main()
9      {
10         var html =
11         @"<TD class=texte width=""50%"">
12             <DIV align=right>Name :<B> </B></DIV>
13         </TD>
14         <TD width=""50%"">
15             <INPUT class=box value=John maxLength=16 size=16 name=user_name>
16             <INPUT class=box value=Tony maxLength=16 size=16 name=user_name>
17             <INPUT class=box value=Jams maxLength=16 size=16 name=user_name>
18         </TD>
19         <TR vAlign=center>";
20
21         var htmlDoc = new HtmlDocument();
22         htmlDoc.LoadHtml(html);
23
24         var htmlNodes = htmlDoc.DocumentNode.SelectNodes("//td/input");
25
26         foreach (var node in htmlNodes)
27         {
28             Console.WriteLine(node.Attributes["value"].Value);
29         }
30     }
31 }
```

```
John
Tony
Jams
```

Gambar. 5. Penggunaan HTML Agility PackStyle

Pada Gambar 5 merupakan contoh penggunaan dari HTML Agility Pack dengan menggunakan fungsi SelectNodes() dari HTML Selectors. Struktur HTML ditampung pada sebuah variabel bernama html (terletak pada baris ke-10 hingga 19). Baris ke-21 dan 22 digunakan untuk memuat (*load*) struktur HTML dari variabel html. Penggunaan dari fungsi SelectNodes() dapat dilihat pada baris ke-24 dengan paramater ekspresi X-Path adalah "//td/input" untuk memilih semua tag atau node <input> yang terdapat di dalam tag atau node <input>. Baris ke-28 digunakan untuk mencetak hasil dari pembacaan nilai dari atribut "value" yang terdapat pada node yang terpilih pada ekspresi X-Path yang telah ditentukan sebelumnya.

Tag atau node <input> yang terdapat di dalam tag atau node <td> adalah sebanyak 3 buah, sehingga output yang akan dihasilkan (pada baris ke-28) ke layar adalah sebanyak 3 buah juga dengan hasil yang ditampilkan adalah "John", "Tony", dan "Jams".

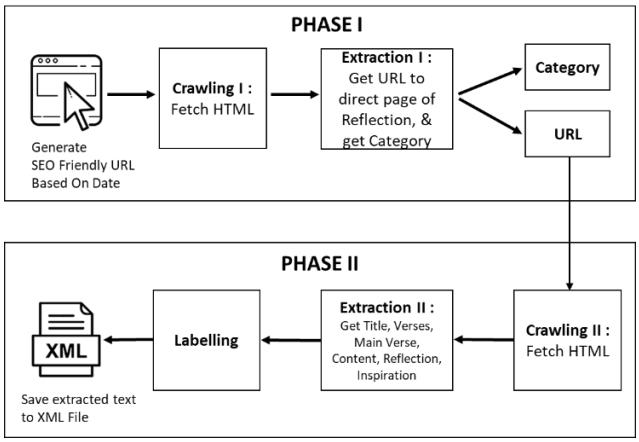
Dengan bantuan HTML Agility Pack untuk melakukan ekstraksi teks dan melalui preproses teks, konten-konten utama yang dibutuhkan dari halaman renungan harian Santapan Rohani dapat dilakukan dengan baik. Dalam penelitian ini, HTML Agility Pack diimplementasikan pada Microsoft Visual Studio 2010, .NET Framework 4.0, dengan bahasa pemrograman Visual Basic (VB .Net).

III. PERANCANGAN SISTEM

Proses ekstraksi teks atau konten yang dibutuhkan dari setiap renungan harian yang terdapat pada website Santapan Rohani terbagi menjadi 2 fase. Pada Gambar 6 menunjukkan rancangan arsitektur yang akan digunakan dalam melakukan ekstraksi teks pada website Santapan Rohani. Fase Pertama dimulai dari menghasilkan bentuk *SEO Friendly URL* untuk melakukan proses *fetching/crawling* HTML ke halaman renungan harian berdasarkan tanggal tertentu pada website Santapan Rohani. Setelah mendapatkan struktur HTML dari proses *fetching/crawling*, kemudian dilakukan ekstraksi teks menggunakan HTML Agility Pack untuk mendapatkan Kategori dan URL.

URL yang didapatkan pada Fase Pertama akan digunakan untuk melakukan proses *crawling/fetching* sekali lagi (*Crawling II)*. Proses ini menjadi penanda dimulainya Fase Kedua. HTML yang didapatkan dari hasil *crawling/fetching* ini kemudian dilakukan ekstraksi konten pada bagian-bagian yang dibutuhkan, seperti: Judul, Ayat Bacaan, Ayat Emas, Isi Renungan, Refleksi, Kutipan Inspirasi. Setelah melakukan ekstraksi maka hasil tersebut disimpan ke dalam file XML agar dapat digunakan lebih lanjut bagi penelitian selanjutnya.

Selain menggunakan HTML Agility Pack untuk melakukan ekstraksi teks pada HTML, setiap proses ekstraksi yang dilakukan pada kedua fase juga melibatkan preproses teks untuk mendapatkan hasil yang maksimal.



Gambar. 6. Rancangan Arsitektur Sistemrata

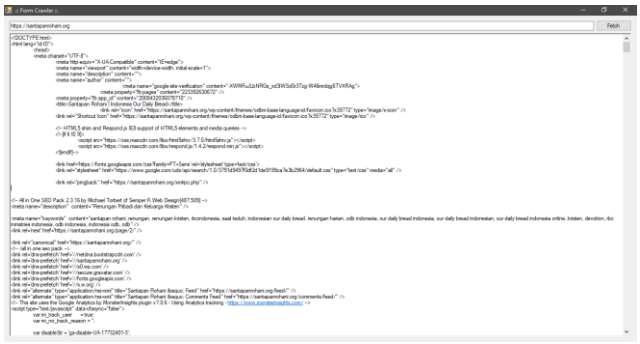
IV. HASIL DAN PEMBAHASAN

Setiap fase dalam penelitian ini terdapat proses *crawling* atau *fetching* yang diperlukan untuk mendapatkan struktur HTML dari suatu halaman website. Pada Segmen Program 3 merupakan penulisan program yang disesuaikan ke dalam bahasa pemrograman VB .Net untuk mendapatkan struktur HTML dari suatu halaman website dengan menggunakan *WebRequest Class* [10]. Baris ke-4 adalah modifikasi dengan menambahkan 1 baris perintah program agar pesan *error* "*The request was aborted: Could not create SSL/TLS secure channel.*" dapat ditangani ketika melakukan *fetching* dari halaman suatu website yang berada pada protokol HTTPS (port 443).

Segmen Program 3. Potongan Program untuk Fetching HTML

```
01. Function fetch(ByVal URL As String, Optional
    ByVal iTimeout As Integer = 10000)
02.   Dim responseFromServer As String = ""
03.   Try
04.     ServicePointManager.SecurityProtocol       =
          CType(192,    SecurityProtocolType)    Or
          CType(768,    SecurityProtocolType)    Or
          CType(3072,SecurityProtocolType)
05.     Dim     request     As     WebRequest     =
          WebRequest.Create(URL)
06.     Dim    response    As    HttpWebResponse    =
          CType(request.GetResponse(),
          HttpWebResponse)
07.     Dim     dataStream     As     Stream     =
          response.GetResponseStream()
08.     Dim reader As New StreamReader(dataStream)
09.     responseFromServer = reader.ReadToEnd()
10.     reader.Close()
11.     dataStream.Close()
12.     response.Close()
13.   Catch Ex As Exception
14.     Dim errorForm As New frmError
15.     errorForm.tbError.Text = Ex.Message
16.     errorForm.ShowDialog()
17.   End Try
18.   Return responseFromServer
19. End Function
```

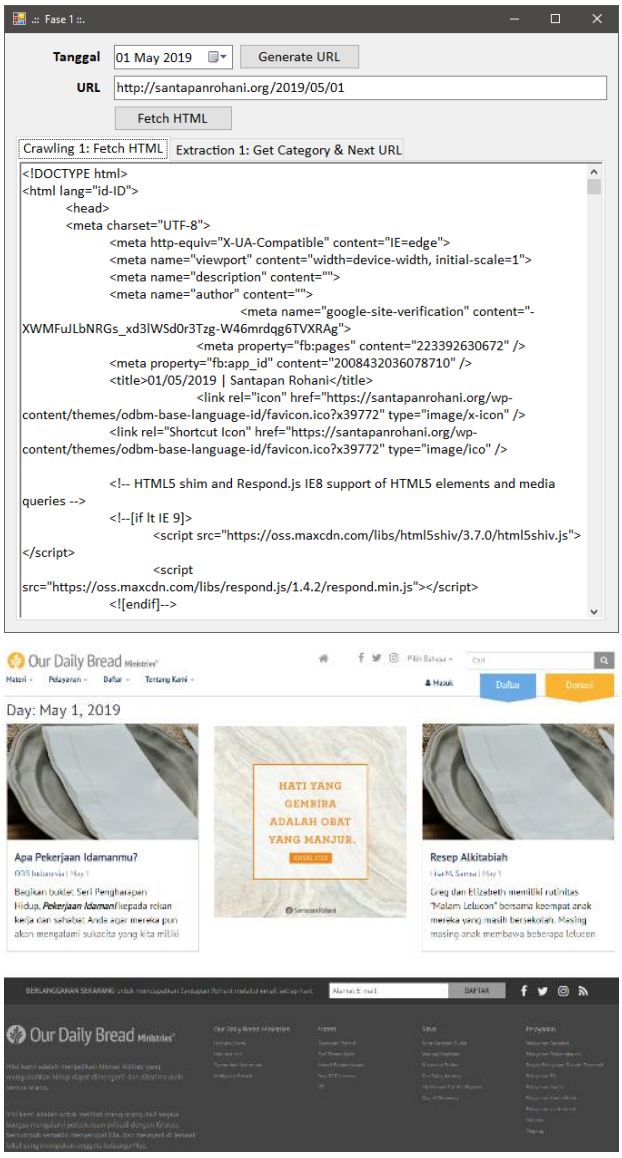
Jika *fetching* HTML tidak berhasil maka akan memberikan hasil *string* kosong (“”), sedangkan jika berhasil akan memberikan hasil dalam bentuk teks HTML dari suatu halaman website seperti yang ditunjukkan pada Gambar 7. Hasil dari proses *fetching* HTML kemudian digunakan sebagai data input untuk proses selanjutnya. Proses selanjutnya adalah melakukan ekstraksi teks-teks yang dibutuhkan dari struktur HTML yang telah didapatkan. Dalam proses ekstraksi teks inilah dibutuhkan bantuan *library* HTML Agility Pack berdasarkan ekspresi X-Path. Setiap bagian teks yang dibutuhkan dari halaman renungan harian pada Website Santapan Rohani menggunakan ekspresi X-Path tertentu.



Gambar. 7. Hasil Uji Coba *Fetching* HTML

Seperti yang telah dijelaskan pada bagian perancangan sistem, Fase Pertama dimulai dengan melakukan generasi URL untuk mengunjungi halaman website Santapan Rohani yang diikuti dengan format tanggal, bulan, dan tanggal seperti yang terlihat pada Gambar 8. Sebagai uji coba dari Fase Pertama dilakukan *fetching* HTML untuk tanggal 1 Mei 2019. Hasil generasi SEO *Friendly URL* yang akan dituju pada contoh ini adalah https://santapanrohani.org/2019/05/01. URL yang dihasilkan akan digunakan melakukan *fetching* HTML dengan menggunakan fungsi yang dikembangkan seperti pada Segmen Program 3 dan hasil *fetching* HTML dapat dilihat pada Gambar 8. Pada Gambar 8 yang terletak di bagian B merupakan bentuk visual dari struktur HTML yang berhasil didapatkan pada *form* Fase 1 (bagian A).

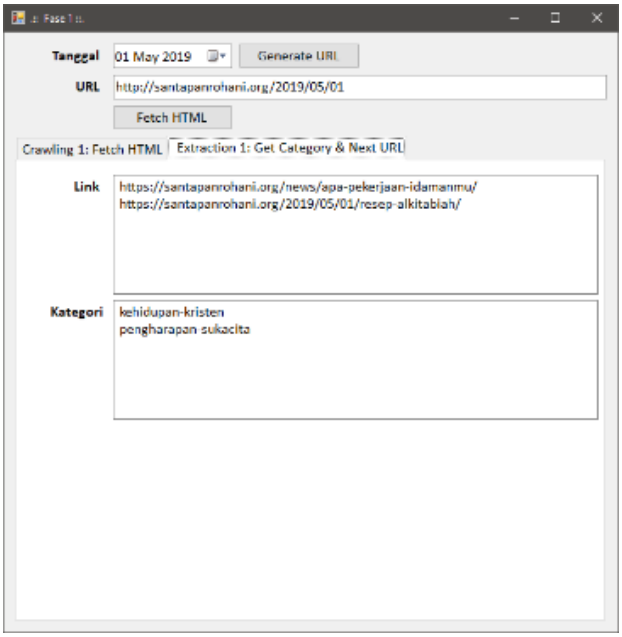
Dari hasil *fetching* URL, fokus dari Fase Pertama ini adalah untuk melakukan ekstraksi teks berupa kategori renungan harian dan URL atau link berikutnya yang akan digunakan pada proses *crawling* atau *fetching* HTML (Fase Kedua) seperti yang terlihat pada Gambar 9. Pada hasil uji coba ekstraksi, terdapat URL atau Link yang akan digunakan (dapat dilihat pada Gambar 9). Hal ini dikarenakan sewaktu ekstraksi URL dengan menggunakan ekspresi X-Path “//h3/a[@href]” terdeteksi oleh HTML Agility Pack ada dua URL atau Link yang memiliki ekspresi X-Path tersebut, sehingga diperlukan preproses teks lebih lanjut agar hasil yang didapatkan lebih akurat.



Gambar. 8. Hasil Fetching HTML pada Fase Pertama

Output URL yang akan digunakan untuk Fase Kedua adalah URL yang hanya mengandung bentuk URL seperti hasil generasi SEO *Friendly URL*.

Setiap renungan harian yang dimiliki oleh memiliki kategori atau topik renungan. Namun, pengelompokkan kategori atau topik renungan hanya dapat ditemui versi cetak. Dalam pengamatan peneliti, kategori atau topik renungan dapat ditemukan pada struktur HTML dari Fase Pertama. Kategori atau topik renungan tidak dapat dilihat langsung pada bentuk visual dari struktur HTML, sehingga perlu dilakukan usaha lebih untuk mendapatkan ekstraksi topik atau kategori renungan dari struktur HTML yang ada. Ekstraksi teks untuk "Kategori Renungan" terdapat pada *class* tag (baris ke-4) yang menjadi *parent* dari tag <a> yang berisikan URL yang dibutuhkan untuk untuk Fase Kedua (dapat dilihat pada ilustrasi struktur HTML Fase Pertama, Gambar 9).



Gambar. 9. Output dari Fase Pertama

Ekstraksi kategori yang didapatkan masih mengandung "-", sehingga dilakukan juga preproses teks agar karakter "-" diubah menjadi " " (spasi), dan mengganti penulisan yang semula *lowercase* menjadi *camelcase*, contoh: dari "kehidupan-kristen" diubah menjadi "Kehidupan Kristen".

Segmen Program 4. Ilustrasi Struktur HTML Fase Pertama

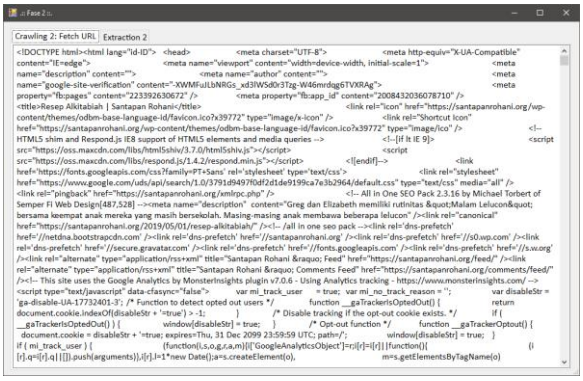
```
01. <div id="content" class="row entries grid-view">
02.   <div class='entry'>...</div>
03.   <div class='entry'>...</div>
04.   <div clss='entry category-...>
05.     <h3>
06.       <a href='...'>...</a>
07.     </h3>
08.      ...
09.   <div>
10. </div>
```

Pada Tabel I merupakan ekspresi X-Path yang digunakan ke dalam HTML Agility Pack pada proses ekstraksi teks untuk mendapatkan Kategori renungan harian dan URL atau link selanjutnya (Fase Pertama).

TABEL I
EKSPRESI X-PATH FASE PERTAMA

Ekstraksi	Ekspresi X-Path
Kategori	//div[@id='content']/div
URL atau Link	//h3/a[@href]

Output URL yang didapatkan pada Fase Pertama digunakan sebagai inputan untuk memulai proses *fecthing* atau *crawling* pada Fase Kedua. Hasil dari *fetching* HTML pada Fase Kedua ditunjukkan pada Gambar 10.



Gambar. 10. Hasil Fetching HTML pada Fase Kedua

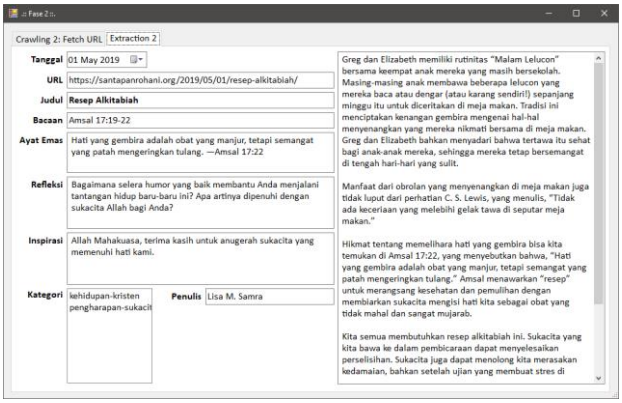
Pada Tabel II merupakan ekspresi X-Path yang digunakan ke dalam HTML Agility Pack pada proses ekstraksi teks untuk mendapatkan Judul, Ayat Bacaan, Ayat Emas, Isi Renungan, Refleksi, Kutipan Inspirasi dan Penulis Renungan.

TABEL II
EKSPRESI X-PATH FASE KEDUA

Ekstraksi	Ekspresi X-Path
Judul	//h2[@class='entry-title']
Ayat Bacaan	//div[@class='passage-box']//span//a
Ayat Emas	//div[@class='verse-box']
Isi Renungan	//div[@class='post-content']//p
Refleksi	//div[@class='poem-box']
Kutipan Inspirasi	//div[@class='thought-box']
Penulis	//span[@class='vcard']//a

Struktur HTML yang digunakan pada Fase Kedua dapat dilihat pada Segmen Program 2. Hasil ekstraksi teks yang dibutuhkan dari struktur HTML yang didapatkan pada Fase Kedua dapat dilihat pada Gambar 11.

Pengujian dari sistem Ekstraksi Teks pada Halaman Website salah satu penerbit renungan harian bernama Santapan Rohani ini dilakukan dengan mengambil (*fetching*) data renungan harian selama 3 bulan, yaitu Maret s/d Mei 2019. Sistem ini berhasil digunakan untuk melakukan pengambilan data renungan harian dan melakukan ekstraksi teks yang dibutuhkan dari masing-masing renungan harian yang didapatkan. Ekstraksi teks yang berhasil didapatkan adalah URL Renungan Harian, Judul Renungan, Ayat Bacaan Renungan Harian, Ayat Emas Renungan, Isi Renungan, Refleksi, Kutipan Inspirasi, Kategori atau Topik Renungan, dan Penulis Renungan.



Gambar. 11. Output dari Fase Kedua

Hasil ekstraksi teks yang dibutuhkan kemudian disimpan ke dalam sebuah file XML (Segmen Program 5) agar hasilnya dapat digunakan lagi untuk pengembangan penelitian selanjutnya. Oleh karena itu, penelitian ini menjadi dasar bagi pengembangan penelitian-penelitian lanjutan yang berhubungan dengan *text mining*.

Segmen Program 5. Ilustrasi File XML

```
<?xml version="1.0" encoding="utf-8" standalone
="yes"?>
<data>
  <date>05/01/2019</date>
  <url>...</url>
  <title>Resep Alkitabiah</title>
  <author>Lisa M. Samra</author>
  <mainverse>...</mainverse>
  <content>
    ...
  </content>
  <poem>...</poem>
  <thought>...</thought>
  <category>Kehidupan Kristen, Pengharapan
  Sukacita</category>
</data>
```

V. KESIMPULAN

Melalui penelitian yang dilakukan ini maka dapat disimpulkan bahwa teks atau informasi yang terdapat pada halaman website Santapan Rohani yang berupa bacaan renungan harian dapat diekstrak dengan baik. Hasil ekstraksi teks berhasil mendapatkan berbagai informasi yang dibutuhkan, antara lain: kategori atau topik dari renungan harian, judul renungan harian, ayat bacaan yang digunakan untuk renungan, ayat emas atau utama yang menjadi sorotan, isi renungan, refleksi, dan kutipan yang menginspirasi.

Implementasi ekstraksi informasi dilakukan dengan menggunakan bantuan HTML Agility Pack. Infomasi-informasi yang didapatkan dari hasil ekstraksi kemudian diberi label (berbentuk XML) untuk mempermudah penelitian selanjutnya yang dapat dikembangkan dari penelitian ini. Penelitian selanjutnya yang dapat dikembangkan adalah ekstraksi kata kunci atau fitur, pengklasifikasian, sistem temu balik, dan penelitian-penelitian yang berhubungan dengan *text mining*.

DAFTAR PUSTAKA

[1] "Kisah Kami | Santapan Rohani." [Online]. Available: https://santapanrohani.org/our-story/. [Accessed: 20-Jan-2019].
[2] "URLs." [Online]. Available: https://moz.com/learn/seo/url. [Accessed: 27-Jan-2019].
[3] F. A. Sutanto, "Implementasi Search Engine Optimization (SEO) on Page pada Web UMKM Batik dan Handicraft," pp. 978–979, 2015.
[4] C. Boyd, "The Ultimate Guide for an SEO-Friendly URL Structure," 2017. [Online]. Available: https://www.searchenginejournal.com/seo-friendly-url-structure-2/202790/. [Accessed: 10-Jan-2019].
[5] "What is HTML?" [Online]. Available: https://www.w3schools.com/whatis/whatis_html.asp. [Accessed: 20-Jan-2019].
[6] "Document Object Model (DOM)." [Online]. Available: https://www.w3.org/DOM/. [Accessed: 20-Jan-2019].
[7] "JavaScript HTML DOM." [Online]. Available: https://www.w3schools.com/js/js_htmldom.asp. [Accessed: 20-Jan-2019].
[8] "HtmlAgilityPack." [Online]. Available: https://www.nuget.org/packages/HtmlAgilityPack/. [Accessed: 07-Jan-2019].
[9] M. T. Mahmoudi, "Automatic Creation of Semantic Schema for Accurate Retrieving of Education-Supportive Documents," pp. 28–33, 2012.
[10] "WebRequest Class." [Online]. Available: https://docs.microsoft.com/en-us/dotnet/api/system.net.webrequest?view=netframework-4.0. [Accessed: 18-Jan-2019].

James Wijaya lahir di Ujung Pandang, Sulawesi Selatan, Indonesia, pada tahun 1990. Dia menyelesaikan studi S1 di program studi Sistem Informasi Universitas Pelita Harapan Surabaya pada tahun 2012. James menyelesaikan studi masternya pada program studi S2 Teknologi Informasi, Sekolah Tinggi Teknik Surabaya (STTS). Minat penelitiannya adalah pengembangan aplikasi web dan *text mining*.